

# API - Automation

## Test: Using Postman and CSV file.

### Instructions:

You need to copy the test script into your Postman project and modify it according to the following explanation:

Code block

Please locate this code snippet.

```
1  const config = {
2    ignoredFields: [],      // Fields to ignore during the comparison
3    ignoredPaths: [],      // Specific paths to ignore
4    ignoredFieldTypes: [], // Field types to ignore
5    fieldsToClean: [],     // Fields to clean dynamic content before comparison
6  };
```

### 1. ignoredFields

#### What is it used for?

"A list of field names to ignore during the comparison between the expected response and the actual response."

#### When to use:

"When there are fields containing dynamic or irrelevant data for the test, such as unique identifiers, dates, or tokens that change with each request."

#### Example:

"If the response contains a field like "ConfirmationNumber" that changes each time, you can define:"

```
1  ignoredFields: ["ConfirmationNumber"],
```

"So that the program ignores the value of this field during the comparison."

### 2. IgnoredPaths

#### What is it used for?

"A list of specific paths within the JSON structure to ignore."

#### When to use:

"When you want to ignore a specific field in a particular location within the data structure, especially if the same field name appears in multiple places but you want to ignore only a specific occurrence."

#### Example:

"If, for example, you want to ignore the 'status' field only within the 'metadata' object:"

```
1  ignoredPaths: ["metadata.status"],
```

"So that the program ignores only this field in that specific location."



### 3. ignoredFieldTypes

#### What is it used for?

"A list of field data types to ignore during the comparison."

#### When to use:

"When you want to ignore all fields of a certain data type, such as all numbers or all arrays, especially if this data type contains dynamic values that are not relevant to the test."

#### Example:

If you want the program to ignore all fields that are numbers:

Code block

```
1 ignoredFieldTypes: ["number"],
```

"So that no comparison is performed on the values of fields that are numbers."

### 4. fieldsToClean

In addition to the code snippet we have learned so far, we will need to use another code snippet.

#### The relevant part for us from it:

#### What is it used for?

"A list of field names where dynamic content should be removed before the comparison."

Code block

```
1 fieldsToClean: ["StatusDesc"],
```

And then: error\_id

#### When to use:

"When a specific field contains a combination of static and dynamic content, and you want to compare only the static part. This is particularly useful for messages that include identifiers or codes that change."

#### Example:

If the field contains a message like: "StatusDesc": "The data was.etc. , error\_id: 12345, other..." and you want to ignore the value of error\_id but ensure the message itself is correct:

Code block

```
1 const patterns = [  
2   /(error_id:\s*)([^\s,]+)/g,  
3 ];
```

Code block

```
1 function removeDynamicContent(value, fieldPath) {  
2   const fieldName = fieldPath.split('.').pop();  
3   if (typeof value === 'string' && config.fieldsToClean  
4     .includes(fieldName)) {  
5     // Patterns to match and remove dynamic content  
6     const patterns = [  
7       // Match 'error_id: dynamic_value' and remove the  
8       // dynamic value, keeping 'error_id:'  
9       /(error_id:\s*)([^\s,]+)/g,  
10    ];  
11    patterns.forEach(pattern => {  
12      value = value.replace(pattern, '$1'); // Keep  
13      'error_id:', remove dynamic value  
14    });  
15    value = value.trim(); // Remove any trailing spaces  
16    return value;  
17  }
```

Code block

```
1 const patterns = [  
2   // Match 'error_id: dynamic_value' and remove the dynamic  
3   // value, keeping 'error_id:'  
4   /(error_id:\s*)([^\s,]+)/g,  
5 ];
```